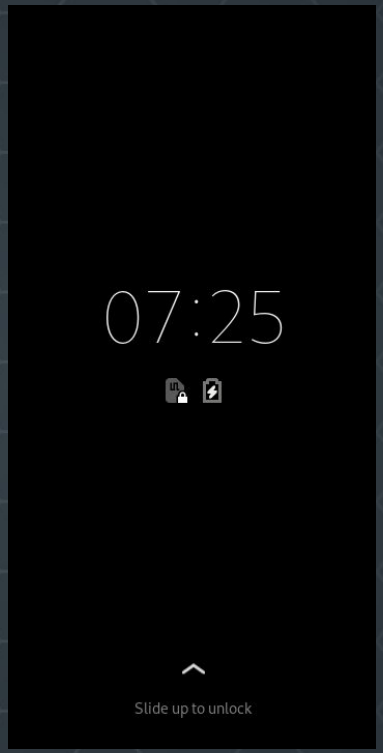


Phosh: A GNOME based Wayland shell and compositor

Tim Orling, Intel Corporation
Joshua Watt, Garmin

Yocto Project Summit *Virtual*, 2021.11

Agenda

- What is Phosh?
- Screenshots
- What works (and what doesn't)
- Can Phosh replace sato/matchbox?
- Next steps

Abstract

The phosh shell and phoc compositor together are a GNOME based GUI designed to work on a mobile phone. We will describe how to build phosh with the meta-phosh layer, look at what components outside of oe-core are required (mostly from meta-gnome), demonstrate how to create an image with gtk+3 applications similar to core-image-sato, and show how to build a pure GNOME image. Finally, we will discuss the open issues remaining to turn phosh into a sato replacement.

Description

Phosh (PHOne SHell) is a shell and phoc (PHOne Compositor) is a compositor for Wayland. Unlike many other compositors, it is based on GNOME and gtk+3 without any other requirements (such as Qt). Phosh was developed by Purism and is now part of GNOME World and under active development in the GNOME community. Phosh and phoc implement the standard GNOME desktop interfaces, meaning it is capable of running gtk+3 and GNOME applications without modification. This makes it a good candidate for replacing matchbox and sato (core-image-sato). Because phosh was designed for mobile phone use, there are some implementation details that will need to be overcome before it can be considered a full-fledged replacement for sato and matchbox. Examples of these current issues are the default lock screen (where the PIN is numeric only) and default portrait alignment of the display.

Content and Continuous Integration

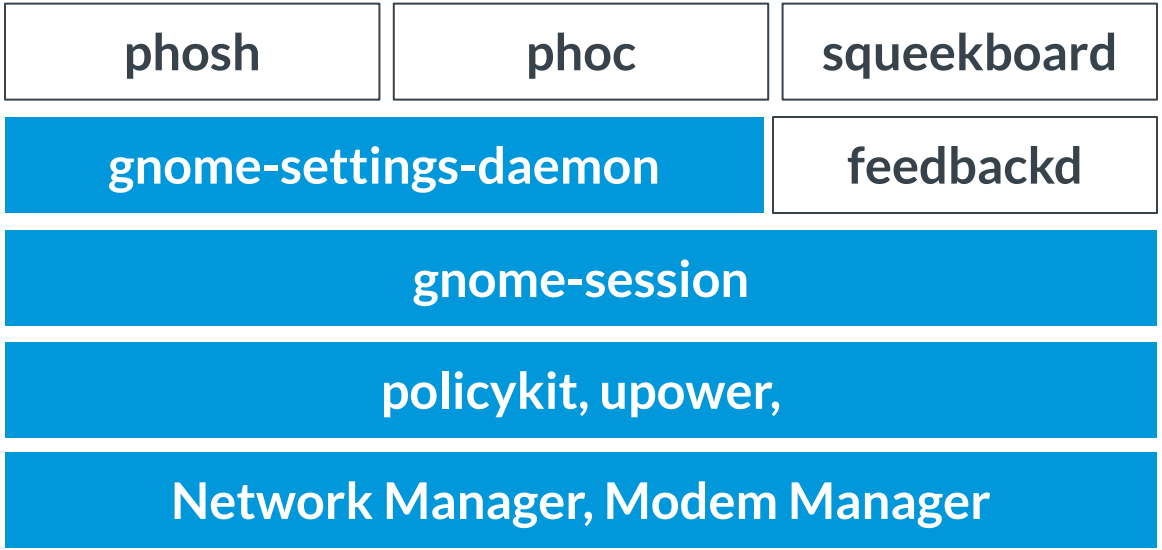
https://github.com/JPEWdev/meta-phosh

What is Phosh?

What is Phosh?

http://honk.sigxcpu.org/con/

- Phone shell (phosh) and Phone compositor (phoc) for Wayland
- Uses GNOME and Gtk APIs; D-Bus



packagegroup-phosh-essential

```
SUMMARY = "Phosh Shell"
PR = "r0"

PACKAGE_ARCH = "${MACHINE_ARCH}"

inherit packagegroup features_check
REQUIRED_DISTRO_FEATURES = "wayland"

RDEPENDS:${PN} = "\
    gnome-control-center \
    phoc \
    phosh \
    squeekboard \
    glibc-localedatas \
    "
```

phosh DEPENDS on feedbackd

core-image-phosh (a sato clone)

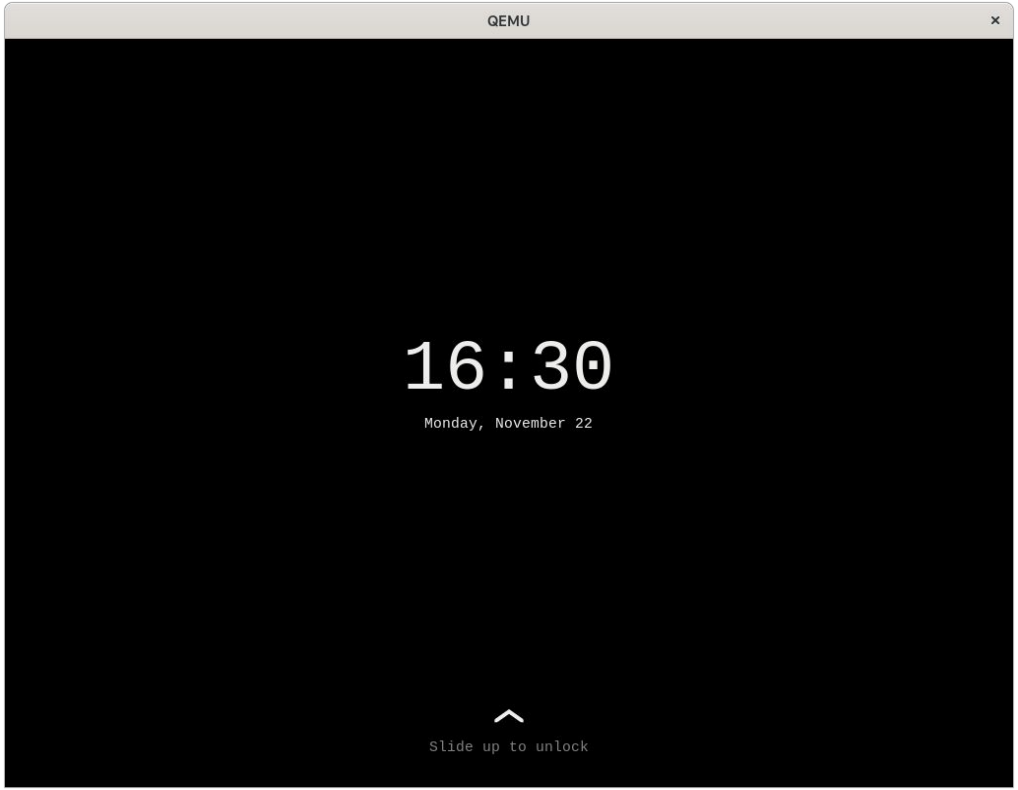
```
SUMMARY = "A basic Phosh image"

IMAGE_FEATURES += "splash package-management ssh-server-dropbear hwcodecs"

LICENSE = "MIT"

inherit core-image

CORE_IMAGE_BASE_INSTALL += "\
    packagegroup-phosh-essential \
    gtk+3-demo \
    epiphany \
    puzzles \
    pcmanfm \
    l3afpad \
    gst-examples \
    "
CORE_IMAGE_BASE_INSTALL += "${@bb.utils.contains('DISTRO_FEATURES', 'x11', 'weston-xwayland
matchbox-terminal', '', d)}"

QB_MEM = "-m 512"
```

core-image-phosh-gnome (pure GNOME apps)

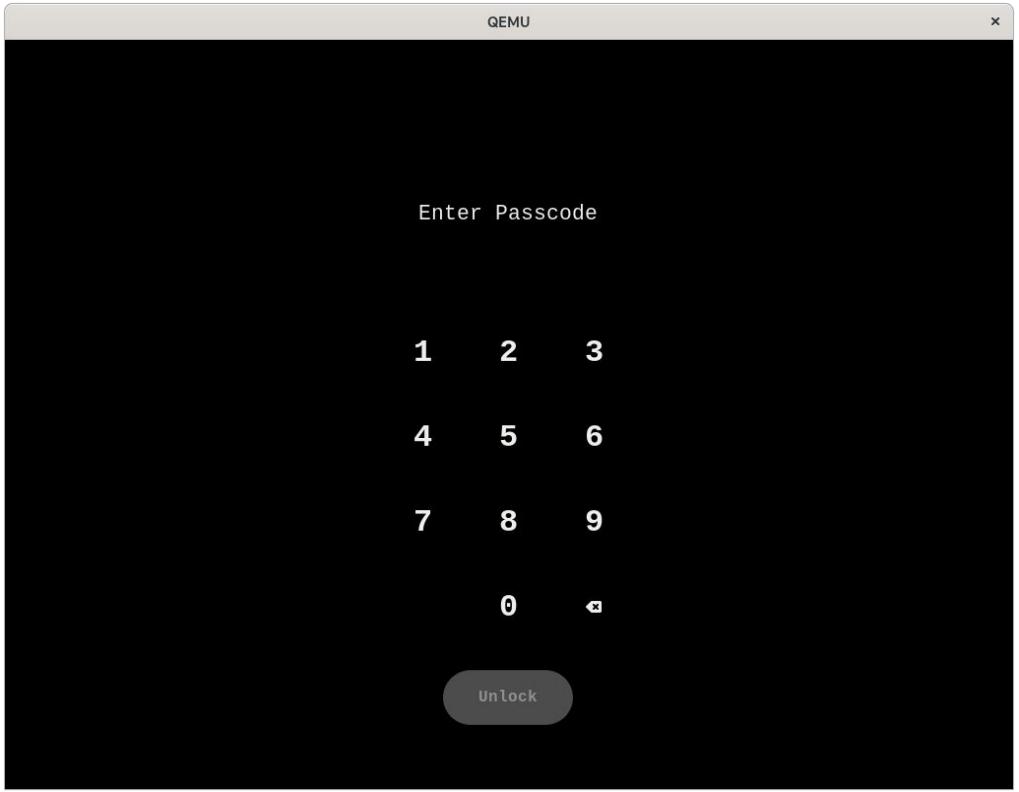
```
SUMMARY = "A Phosh image with GNOME applications"

IMAGE_FEATURES += "splash package-management ssh-server-dropbear hwcodecs"

LICENSE = "MIT"

inherit core-image

CORE_IMAGE_BASE_INSTALL += "\
    packagegroup-phosh-essential \
    epiphany \
    packagegroup-gnome-apps \
    "
CORE_IMAGE_BASE_INSTALL += "${@bb.utils.contains('DISTRO_FEATURES', 'x11',
'weston-xwayland', '', d)}"


QB_MEM = "-m 1024"
```

Screenshots

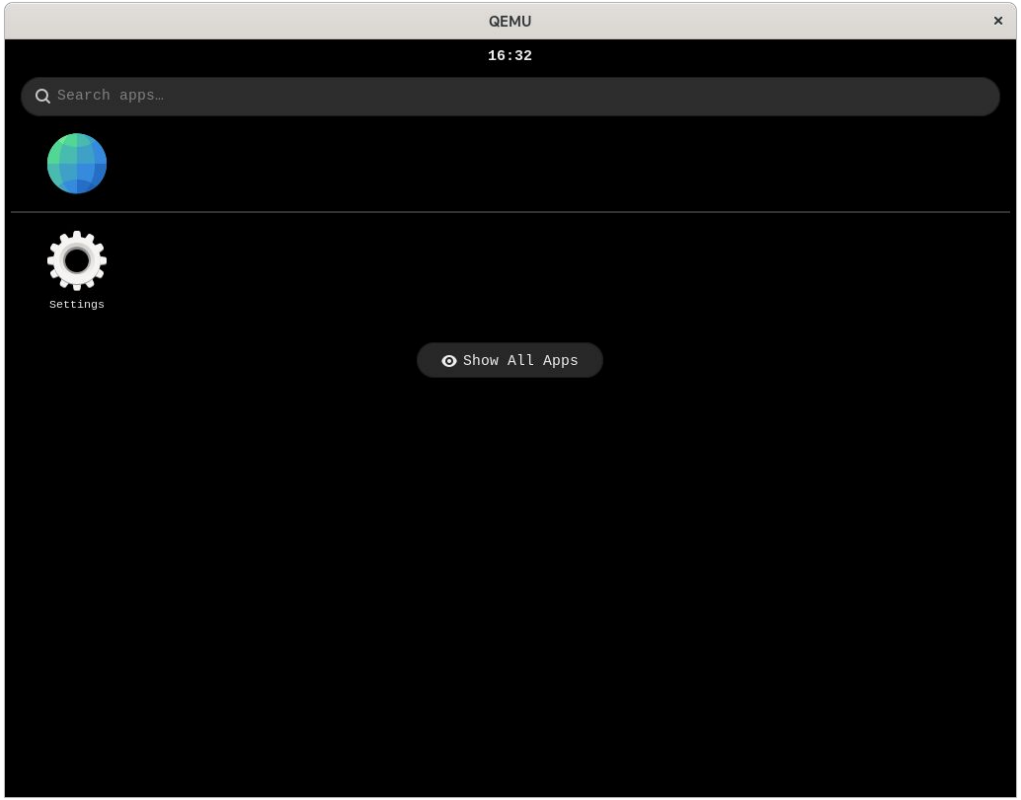
Lock-screen on qemux86-64



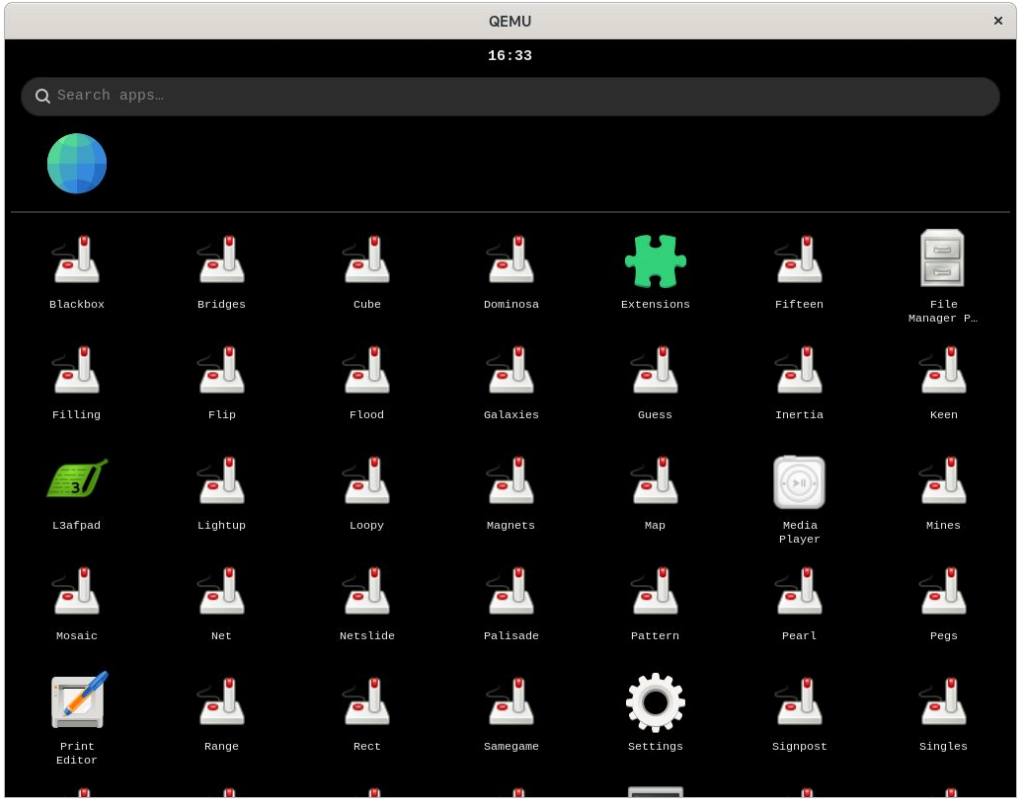
PIN entry screen on qemux86-64



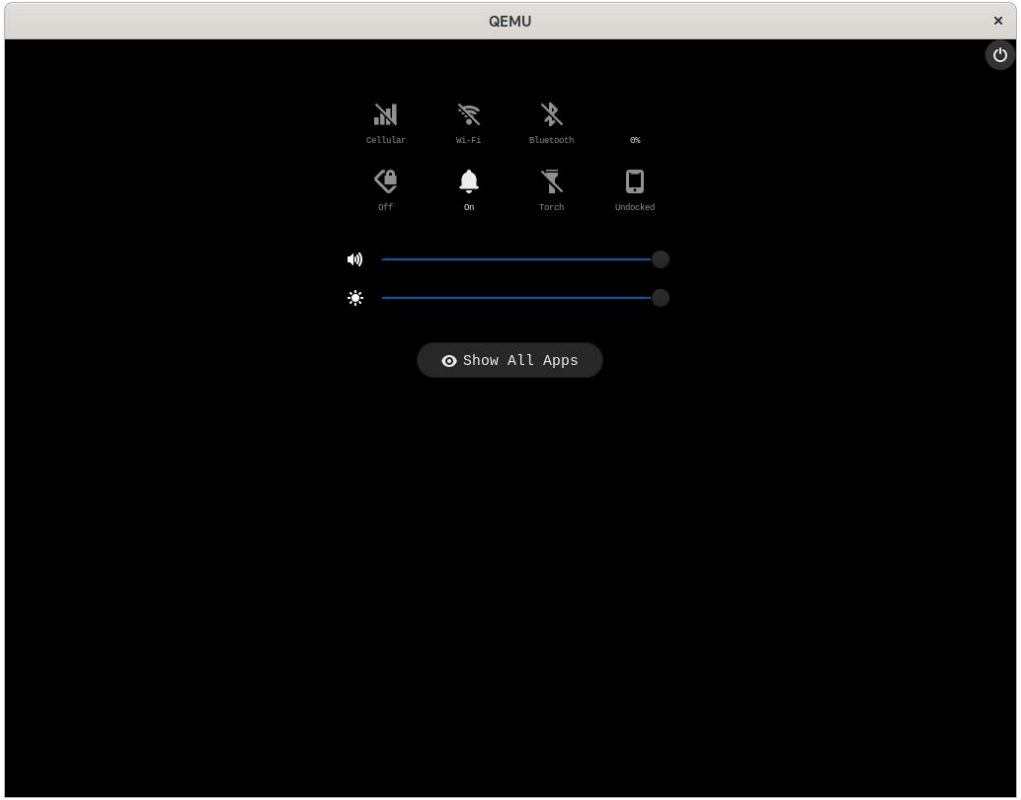
App thumbnail view on qemux86-64



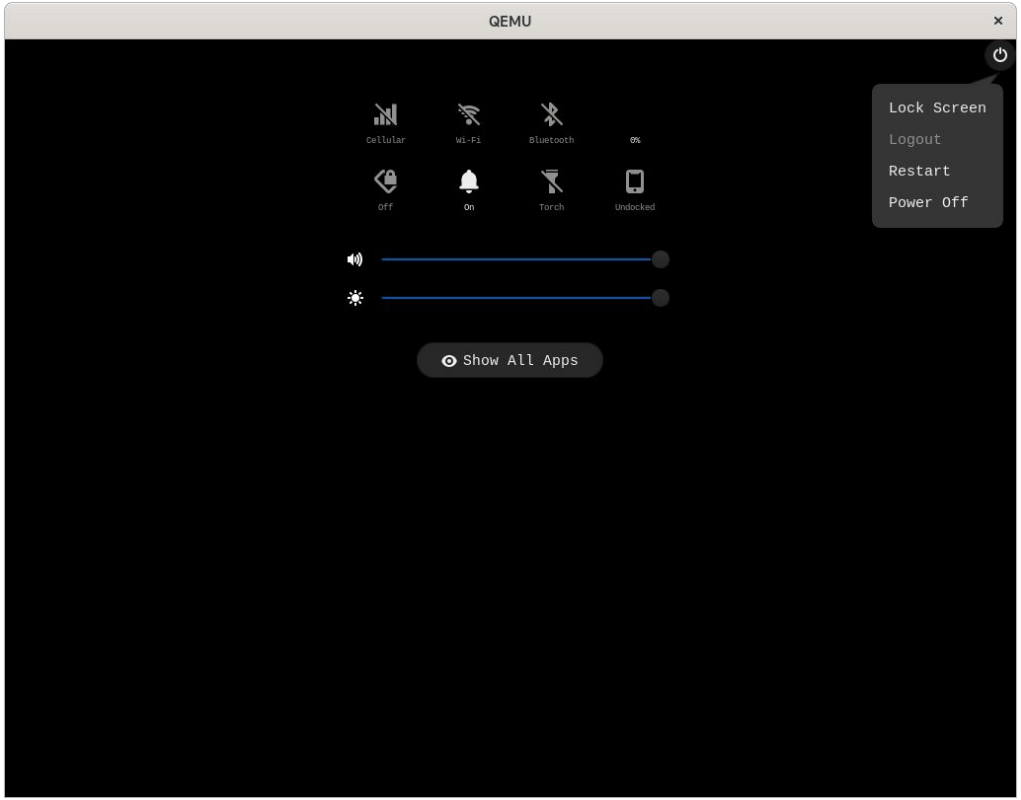
Expanded App thumbnails view on qemux86-64



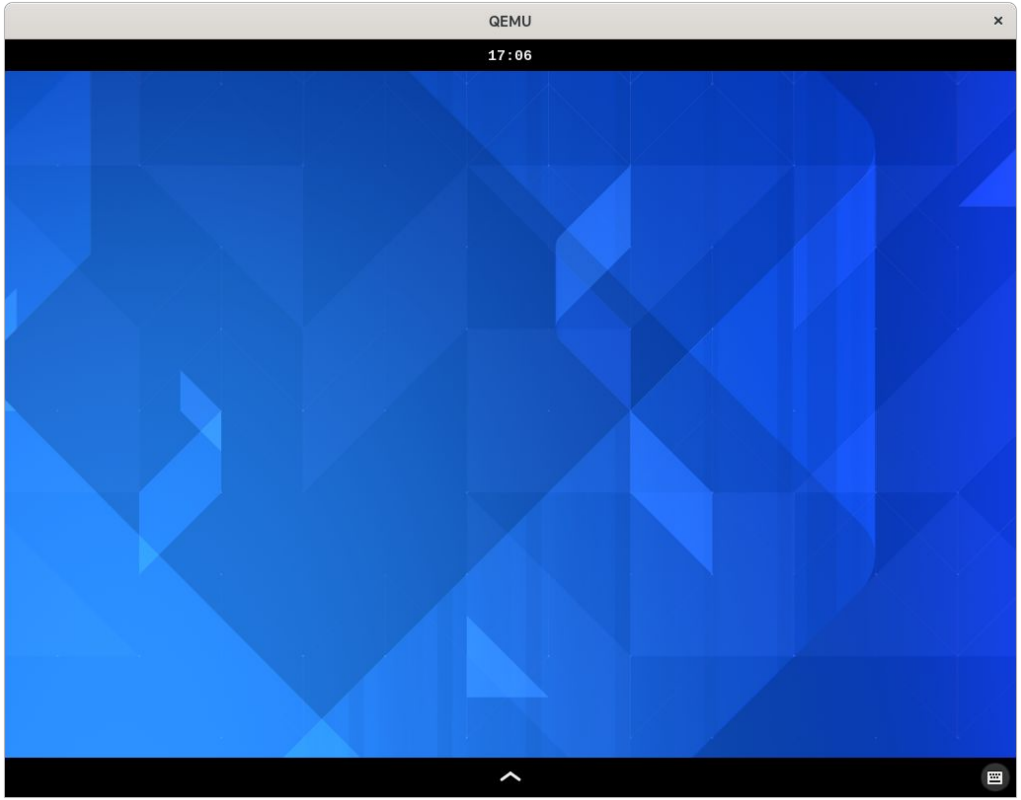
Top panel on qemux86-64



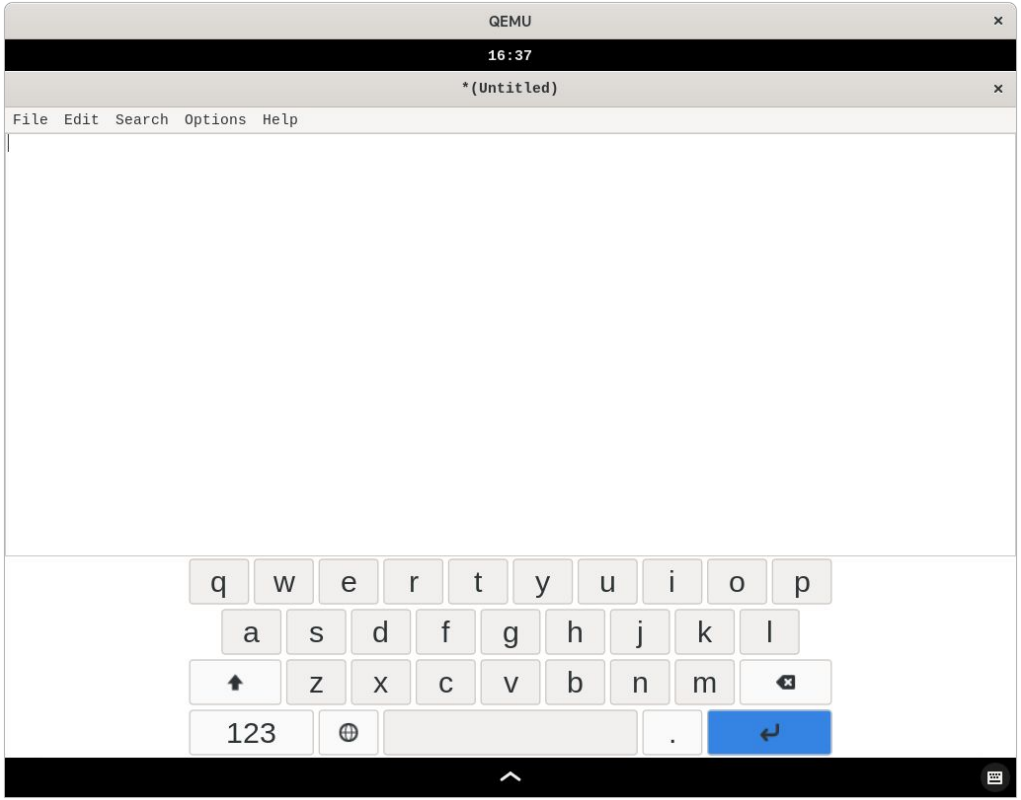
Power menu on qemux86-64



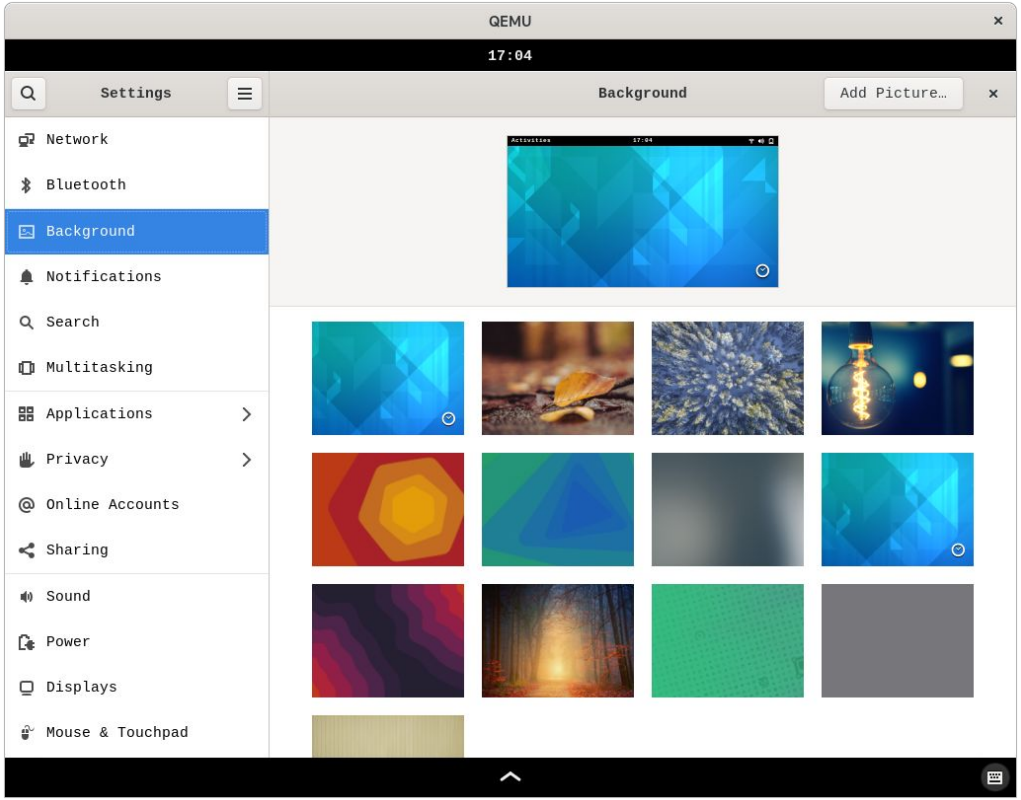
“Desktop” background on qemux86-64



L3afpad with squeekboard on qemux86-64

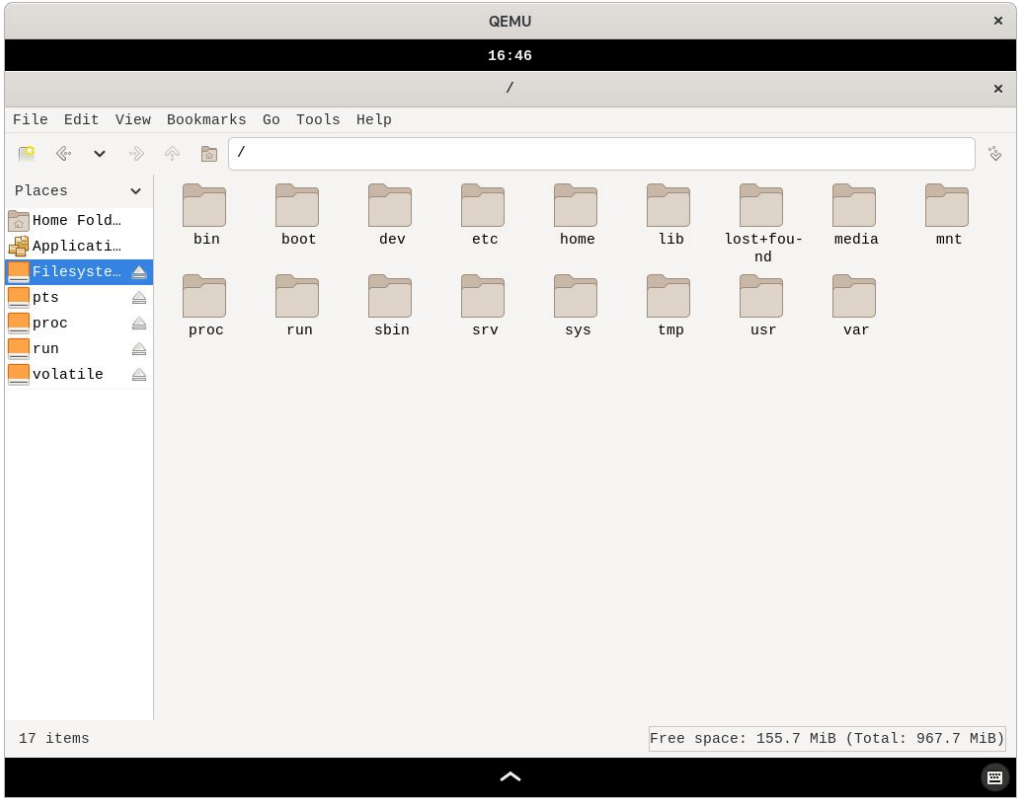


Background settings on qemux86-64

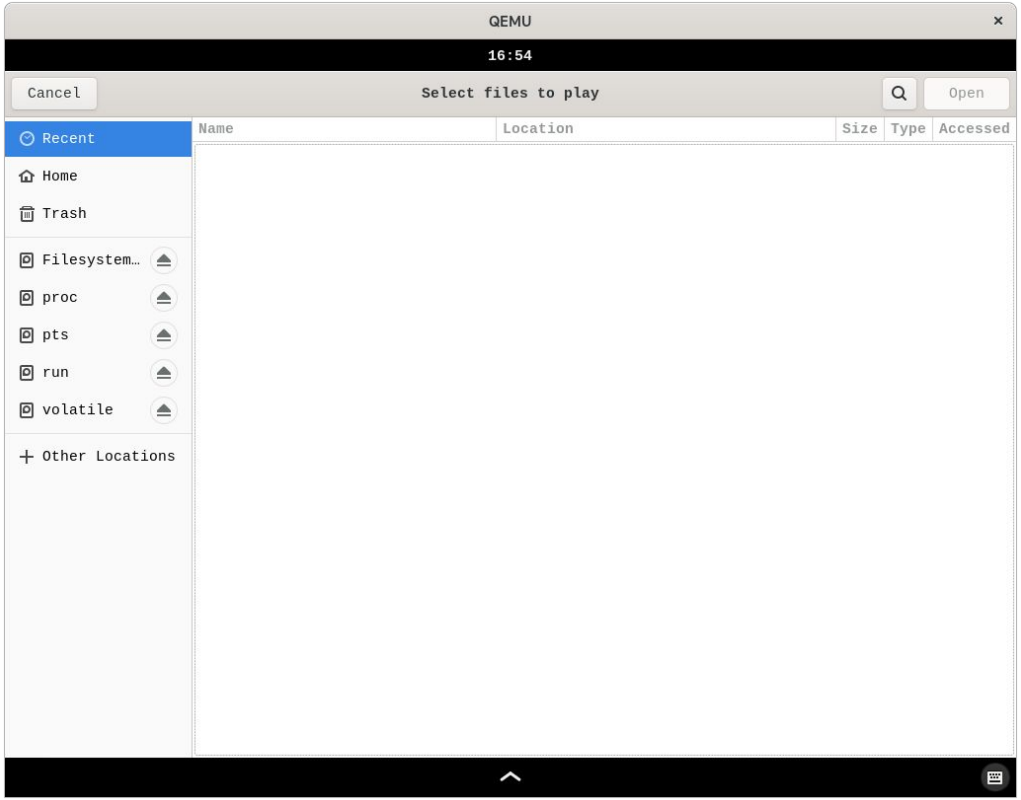


Matchbox terminal on qemux86-64

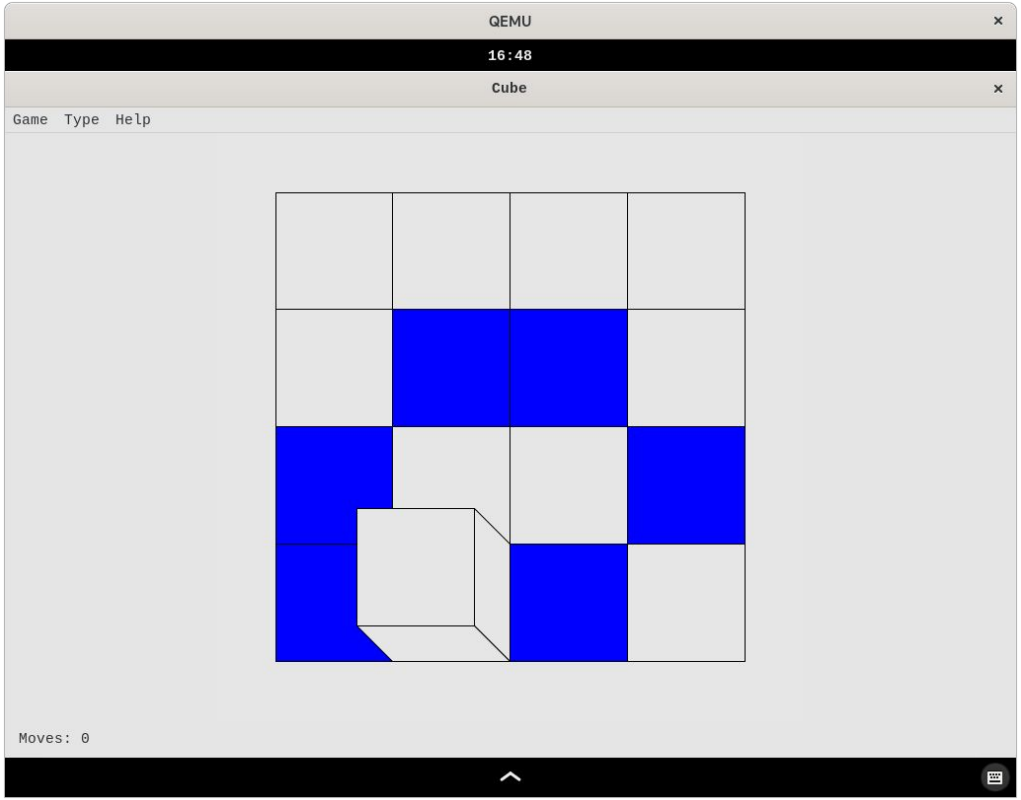
File roller on qemux86-64



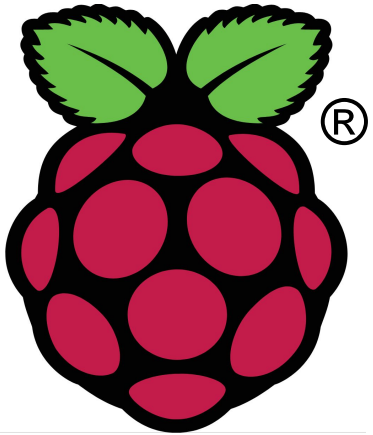
Media player on qemux86-64



puzzles “cube” on qemux86-64



Yes, but can I run it on my



?

YES

What works

and what doesn’t

What works?

- Gnome and GTK(sato/matchbox) apps show on launch screen
- Matchbox terminal works as expected
- Works well on supported GPU hardware (meta-raspberrypi)

...and what doesn't? (not an exhaustive list)

- **gnome-calculator launches full screen**
- **Some apps launch, but then shows background and not app**
  - gnome-terminal (locales issue?)
  - media-player (delayed, eventually shows app)
  - gtk-print-editor (delayed, eventually shows app)
- **mouse cursor is offset in QEMU, awkward experience**

Can Phosh replace sato/matchbox?

Can Phosh replace sato/matchbox?

- Most apps work or almost work
- Does not seem to have concept of multiple tabs of apps
- Some migration from meta-gnome to oe-core

Since it is maintained by upstream Debian and GNOME and others, more likely to stay up to date and add new features/fix bugs compared to our grow-your-own desktop (sato/matchbox).

Next steps

Next steps

- Figure out how to not require PIN to login
- Configuration of GNOME apps to be more “mobile friendly”
- Troubleshoot apps which fail to launch completely

Thank You

- Purism for making this Open Source and upstreaming to https://gitlab.gnome.org/World/Phosh
- All the Open Embedded contributors working on the GNOME stack (oe-core, meta-oe, meta-gnome)
- Alex Kanavin and Ross Burton for encouragement and advice

What is the Yocto Project® ?

IT'S NOT AN EMBEDDED LINUX DISTRIBUTION, IT CREATES A CUSTOM ONE FOR YOU.

The Yocto Project (YP) is an open source collaboration project that helps developers create custom Linux-based systems regardless of the hardware architecture.

The project provides a flexible set of tools and a space where embedded developers worldwide can share technologies, software stacks, configurations, and best practices that can be used to create tailored Linux images for embedded and IOT devices, or anywhere a customized Linux OS is needed.

yocto PROJECT | THE LINUX FOUNDATION